CLAIMS

What is claimed is:

Sub A1

~~1. A collector for collecting non-referenced objects stored in a heap by a program executing in a computer system comprising:~~

~~an object allocation routine which stores an object of a particular type in one of a plurality of spaces in the heap dependent on a predefined category for the type; and~~

~~a collection routine which searches one of the spaces for referenced objects and reclaims non-referenced objects stored in the searched space.~~

Sub B2

~~2. The collector as claimed in Claim 1 further comprising:~~

~~a sample and partition routine which defines a category of an object stored in the heap to be hot or cold.~~

Sub A1

~~3. The collector as claimed in Claim 2 wherein upon determining that hot space is full, the collection routine searches cold space and hot space for referenced objects and moves referenced objects of the hot category stored in hot space to cold space.~~

~~4. The collector as claimed in Claim 2 wherein the sample and partition further comprises:~~

~~a write barrier elimination routine, which eliminates a write barrier for an intergenerational pointer between an object stored in hot space and an object stored in cold space.~~

SUB B47

5. The collector as claimed in Claim 4 wherein the write barrier elimination routine eliminates a write barrier by replacing a write barrier machine code instruction with a no operation machine code instruction.

6. The collector as claimed in Claim 2 wherein the sample and partition routine defines the object category dependent on object type mortality.

7. The collector as claimed in Claim 6 wherein the sample and partition routine estimates the object mortality dependent on difference of the number of bytes of the object type stored in the heap before a collection and the number of bytes of the object type stored in the heap after the collection.

Sub a3

8. The collector as claimed in Claim 2 wherein the sample and partition routine partitions the heap to minimize intergenerational pointers between hot space and cold space.

9. A collector for collecting non-referenced objects stored in a heap by a program executing in a computer system comprising:

means for storing an object of a particular type in one of a plurality of spaces in the heap dependent on a predefined category for the type;

means for searching one of the spaces for referenced objects; and

means for reclaiming non-referenced objects stored in the searched space.

10. The collector as claimed in Claim 9 further comprising:

means for partitioning the heap into cold space and hot space by defining a category of an object stored in the heap to be hot or cold.

Sub a3

~~11. The collector as claimed in Claim 10 wherein upon determining that hot space is full, the means for searching searches cold space and hot space for referenced objects and moves referenced objects stored in the hot space to a cold space.~~

~~12. The collector as claimed in Claim 10 wherein the means for partitioning further comprises:~~

~~means for eliminating a write barrier for an intergenerational pointer between an object stored in hot space and an object stored in cold space.~~

Sub B67

~~13. The collector as claimed in Claim 12 wherein the means for eliminating a write barrier replaces write barrier machine code instructions with no operation machine code instructions.~~

~~14. The collector as claimed in claim 10 wherein the means for partitioning defines a hot object dependent on object type mortality.~~

~~15. The collector as claimed in Claim 14 wherein the means for partitioning estimates the object mortality dependent on difference of the number of bytes of the object type stored in the heap before a collection and the number of bytes of the object type stored in the heap after the collection.~~

Sub a4

~~16. The collector as claimed in Claim 9 wherein the means for partitioning partitions the heap to minimize intergenerational pointers between hot space and cold space.~~

~~17. A method for collecting non-referenced objects stored in a heap by a program executing in a computer system comprising the steps of:~~

~~storing an object of a particular type in one of a plurality of spaces in the heap dependent on a predefined category for the type;~~

Sub a4

~~searching one of the spaces for referenced objects; and~~

~~reclaiming non-referenced objects stored in searched space.~~

~~18. The method as claimed in Claim 17 further comprising the step of:~~

~~partitioning the heap into cold space and hot space by defining hot space objects and cold space objects.~~

~~19. The method as claimed in Claim 18 wherein upon determining that hot space is full, the step of reclaiming further comprises the step of:~~

~~moving referenced objects stored in the hot space to a cold space.~~

~~20. The method as claimed in Claim 18 wherein the step of partitioning further comprises the step of:~~

~~eliminating a write barrier for an intergenerational pointer between an object stored in hot space and an object stored in cold space.~~

Sub B8

~~21. The method as claimed in Claim 20 wherein the step of eliminating a write barrier replaces write barrier machine code instructions with no operation machine code instructions.~~

~~22. The method as claimed in claim 18 wherein the step of partitioning further comprises the step of:~~

~~identifying a hot object dependent on object type mortality.~~

~~23. The method as claimed in Claim 22 wherein the step of identifying estimates the object type mortality dependent on difference of the number of bytes of the object type stored in the heap before a collection and the number of bytes of the object type stored in the heap after the collection.~~

Sub a5

24. The method as claimed in Claim 18 wherein the step of partitioning partitions the heap to minimize intergenerational pointers between hot space and cold space.

25. A computer system comprising:

a central processing unit connected to a memory bus by a system bus;

an I/O system, connected to the system bus by a bus interface; and

a collector for collecting non-referenced objects stored in a heap by a program executing in a computer system, the collector:

storing an object of a particular type in one of a plurality of spaces in the heap dependent on a predefined category for the type;

searching one of the spaces for referenced objects; and

reclaiming non-referenced objects stored in searched space.

26. A computer program product for collecting non-referenced objects stored in a heap by a program executing in a computer system, the computer program product comprising a computer usable medium having computer readable program code thereon, including program code which:

stores an object of a particular type in one of a plurality of spaces in the heap dependent on a predefined category for the type;

searches one of the spaces for referenced objects; and

reclaims non-referenced objects stored in searched space.